

UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA
INGENIERIA EN SISTEMAS DE INFORMACION Y CIENCIAS DE LA
COMPUTACION CAMPUS JUTIAPA



TECER SEMESTRE

SECCION: "A"

- **Robertson Alessandro Muñoz Paredes**

Carnet: 0905-24-16433

Progra 1

Proyecto parcial 2

Cuestionario a responder:

1. En el método Crear de la clase JugadorService, ¿por qué se utiliza SCOPE_IDENTITY() en la consulta SQL y qué beneficio aporta al código?

R// el scope identity sirve para darle un valor de identidad ya se en alguna tabla o en alguna columna

2. En el método Eliminar del servicio de jugadores, ¿por qué se verifica la existencia de elementos en el inventario antes de eliminar un jugador y qué problema está previniendo esta comprobación?

R// en lo que estuve probando el código me percate que se verifica para verificar lo que tenga el usuario en su inventari, y el problema que previene es también a la momento que por accidente eliminemos a un jugador o al inventario de cosas que tiene el

3. ¿Qué ventaja ofrece la línea using var connection = _dbManager.GetConnection(); frente a crear y cerrar la conexión manualmente? Menciona un posible problema que podría ocurrir si no se usara esta estructura.

R// es una ventaja muy común, pero es muy buena ya que es la más recomendada para guarda datos

4. En la clase DatabaseManager, ¿por qué la variable _connectionString está marcada como readonly y qué implicaciones tendría para la seguridad si no tuviera este modificador?

R// esta marcada con un readonly ya que eso ayuda a que no se puedan hacer modificaciones no deseadas, con el readonly se tiene un poco mas de seguridad

5. Si quisieras agregar un sistema de logros para los jugadores, ¿qué cambios realizarías en el modelo de datos actual y qué nuevos métodos deberías implementar en los servicios existentes?

R// el primer cambio lo aria creando una nueva carpeta y en ella hacer una clase nombrada logros y empzar a codificar, ya luego buscaría como modificar la parte de inventario, ya

que sería un logro quedaría aguardado en inventario y también verificaría el apartado de servicios para así poder participar

6. ¿Qué sucede con la conexión a la base de datos cuando ocurre una excepción dentro de un bloque using como el que se utiliza en los métodos del JugadorService?

R// se sierra la base de datos y se liberan los recursos que se tengan asociados

7. En el método ObtenerTodos () del JugadorService, ¿qué ocurre si la consulta SQL no devuelve ningún jugador? ¿Devuelve null o una lista vacía? ¿Por qué crees que se diseñó de esta manera?

R// en la parte donde dice list>jugadores esta una commansql y tiene una conexión y pueda que entre en una lista vacia porque ingresamos mal un comando o también un mal id del jugadores

8. Si necesitaras implementar una funcionalidad para registrar el tiempo jugado por cada jugador, ¿qué cambios harías en la clase Jugador y cómo modificarías los métodos del servicio para mantener actualizada esta información?

R// usaría una funcion que se llama datetime.currentTime para así ver la hora y la fecha en la que empiezan a jugar,

9. En el método TestConnection() de la clase DatabaseManager, ¿qué propósito cumple el bloque try-catch y por qué es importante devolver un valor booleano en lugar de simplemente lanzar la excepción?

R// el try catch es el que maneja las ejecuciones del programa y es importante ya que nos ayuda a que el programa no se dañe, como ejemplo cuando nos indica que pongamos número y ponemos letras el programa se daña

10. Si observas el patrón de diseño utilizado en este proyecto, ¿por qué crees que se separaron las clases en carpetas como Models, Services y Utils? ¿Qué ventajas ofrece esta estructura para el mantenimiento y evolución del proyecto?

R// en mi punto de vista esta más ordenado el código así, ya que cuando alguien más lo quiera manipular ya solo busca la carpeta y la clase en la cual quiere trabajar, y ya no estarán buscando alguna parte del código, y por el cual las carpetas están separadas es según para mí tener un mayor control a la hora que vayamos a copiar el código

11. En la clase InventarioService, cuando se llama el método AgregarItem, ¿por qué es necesario usar una transacción SQL? ¿Qué problemas podría causar si no se implementara una transacción en este caso?

R// en la clase inventario services se llama el método agregaritem ya que es importante usar una transacción sql para garantizar una operación con la base de datos

R// el problema que pueda causar es cuando insertemos otro inventario y quizás se clone y nos de datos incorrectos, también tendría pérdida de datos ya que pueda que busquemos algún bloque y no exista

12. Observa el constructor de JugadorService: ¿Por qué recibe un DatabaseManager como parámetro en lugar de crearlo internamente? ¿Qué patrón de diseño se está aplicando y qué ventajas proporciona?

R// quizás el parámetro de diseño sea por la conexión a sql ya que el en DatabaseManager lo que contiene es una conexión con sql

13. En el método ObtenerPorId de JugadorService, ¿qué ocurre cuando se busca un ID que no existe en la base de datos? ¿Cuál podría ser una forma alternativa de manejar esta situación?

R// como en la base de datos se registran los jugadores y ya tienen su id asignado y buscamos un jugador que nada que ver nos dará un mensaje que dice no se encontró jugador con tal id, lo que se puede hacer al momento es darle al usuario una opción de crear o registrar un jugador al momento que no le aparezcan los jugadores, en esa parte podríamos usar cierta parte que se utilizará para registrar nuevos jugadores

14. Si necesitas implementar un sistema de "amigos" donde los jugadores puedan conectarse entre sí, ¿cómo modificarías el modelo de datos y qué nuevos métodos agregarías a los servicios existentes?

R// como ya se tiene un registro de todos los jugadores, solo crearía una función para reunirlos, y haría un apartado específico para ir llamando uno por uno para que así puedan jugar

15. En la implementación actual del proyecto, ¿cómo se maneja la fecha de creación de un jugador? ¿Se establece desde el código o se delega esta responsabilidad a la base de datos? ¿Cuáles son las ventajas del enfoque utilizado?

R// cuando probe el código y agregaba jugadores, ya cuando lo creaba aparecía una fecha la cual era del día en que se creó ese nuevo jugador, las ventajas de ello es que es mejor para así llevar un control de jugadores nuevos y jugadores veteranos

16. ¿Por qué en el método getConnection () de DriverManager se crea una nueva instancia de SqlConnection cada vez en lugar de reutilizar una conexión existente? ¿Qué implicaciones tendría para el rendimiento y la concurrencia?

R//usando un Getconnection tendríamos una conexión más segura y sqlconnection pueda que se cree cambios en el código y cuando lo copilemos nos dará errores, y con la Getconnection tendría un mejor rendimiento y lo usaríamos en las partes que queremos no solo en una sola parte

17. Cuando se actualiza un recurso en el inventario, ¿qué ocurriría si dos usuarios intentan modificar el mismo recurso simultáneamente? ¿Cómo podrías mejorar el código para manejar este escenario?

R// esta pregunta en mi punto de vista está casi basada con la pregunta 11, ya que si no tenemos la transacción asegurada nos da ese problema que se clone el inventario, y ya cuando los usuarios estén en línea y ambos quieren modificar el recurso les dará el problema ya que se podría decir que se clonaría

18. En el método Actualizar de JugadorService, ¿por qué es importante verificar el valor de rowsAffected después de ejecutar la consulta? ¿Qué información adicional proporciona al usuario?

R//es importante ya que nos ayuda a ver que la operación no tenga ningún fallo, y nos garantiza de que todo salga bien además el rowsaffected también ayuda al usuario a dar una retroalimentación por si no entendiste

19. Si quisieras implementar un sistema de registro (logging) para seguir todas las operaciones realizadas en la base de datos, ¿dónde colocarías este código y cómo lo implementarías para afectar mínimamente la estructura actual?

R// en la carpeta Services pondría la clase llamada repositorio y en ella uno todos ya para que se puedan hacer login, hago un public

DataRepository (ILogger logger)

```
{  
    _logger = logger;  
}
```

Esto ya lo aria como nombrando y ya luego empezar a codificar para que el usuario pueda agregar jugadores o eliminarlos

20. Observa cómo se maneja la relación entre jugadores e inventario en el proyecto. Si necesitaras agregar una nueva entidad "Mundo" donde cada jugador puede existir en múltiples mundos, ¿cómo modificarías el esquema de la base de datos y la estructura del código para implementar esta funcionalidad?

R// para esto crearía una carpeta y en ella pondría la clase mundo y en ella llamaría ambas clases para que se unan y crear diversos portales para que el puedan elegir para los portales pondría un switch con case break para que elijan y les pondría como un tiempo limitado para que puedan ir a otro mundo

21. ¿Qué es un SqlConnection y cómo se usa?

R// el sqlconnection es una funcion que se utiliza cuando tenemos una base de datos y las queremos concetar con nuestro proyecto que llevamos en suiza